AD-A107 070 NAVAL UNDERWATER SYSTEMS CENTER NEWPORT RI F/G 17/1

UNDERWATER TARGET TRACKING USING MEASUREMENTS FROM A LINEAR ARR--ETC(U)

OCT 81

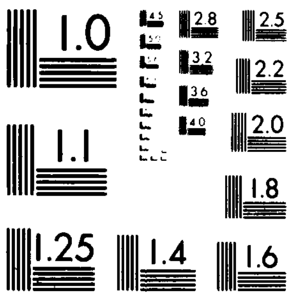
UNCLASSIFIED NUSC-TR-6010 NL

END

DATE FILMED

12-81

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

NUSC Technical Report 6010
22 October 1981

12

Underwater Target Tracking Using Measurements from a Linear Array of Omnidirectional Acoustic Elements

Combat Control Systems Department

Naval Underwater Systems Center
Newport, Rhode Island / New London, Connecticut

PREFACE

This investigation was conducted under NUSC Project No. B45127, "WAA and Multisensor TMA Improvements for Advanced Combat Control Systems," principal investigator, B. W. Guimond (Code 3521), and under Navy Subproject and Task No. S0222-AS. The sponsoring activity is the Naval Sea Systems Command, program manager, R. Cockerill (SEA-63D6).

The major contributor to this report was D. J. Murphy, Professor of Electrical Engineering, Southeastern Massachusetts University, who is a consultant to the Systems Analysis and Synthesis Branch (Code 3521) of the NUSC Combat Control Systems Department. Technical reviewer for this report was B. W. Guimond (Code 3521).

REVIEWED AND APPROVED: 22 October 1981

G. R. Lopriore / for

J. C. Lamb
Head, Combat Control Systems Department

(14) NUSC- (1) Tech. rept.

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
TR-6010

2. GOVT ACCESSION NO.
AD A107070

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
(6) UNDERWATER TARGET TRACKING USING MEASUREMENTS FROM A LINEAR ARRAY OF OMNIDIRECTIONAL ACOUSTIC ELEMENTS.

5. TYPE OF REPORT & PERIOD COVERED

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Combat Control Systems Department

8. CONTRACT OR GRANT NUMBER(s)
(17) S0222AS

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Naval Underwater Systems Center
Newport, Rhode Island 02840

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
Subproject/Task No.
(16) S0222-AS

11. CONTROLLING OFFICE NAME AND ADDRESS
Naval Sea Systems Command (SEA-63D6)
Washington, DC 20632

12. REPORT DATE
(11) 22 Oct[redacted]81

13. NUMBER OF PAGES
66

14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*
(12) 63

15. SECURITY CLASS. *(of this report)*
UNCLASSIFIED

15a. DECLASSIFICATION / DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for Public Release; Distribution Unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*
Underwater Target Tracking
Arrayed Sonar Systems
Acoustic Signal Processing

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

Tracking of underwater targets using measurements from a geometrically linear array of omnidirectional acoustic elements is a nonlinear estimation problem. The technique constructed and used in this report is an iterative, least-squares algorithm which uses the modified Gauss-Newton method. In this algorithm the search direction is found by applying a Householder transformation to the Gauss-Newton equations; the size of the step in this direction is found by minimizing a polynomial approximation to the

DD FORM 1 JAN 73 1473

406068

20. Abstract (Cont'd)

performance index. The results show: (1) the sign on the estimate of the target's depth cannot be uniquely determined, (2) the Gauss-Newton method works as well as its modified version for these experiments, (3) the problem is singular on the first leg, and (4) the initialization of the algorithm influences convergence.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

LIST OF TABLES (Cont'd)

LIST OF SYMBOLS

b = distance between adjacent omnidirectional elements in the linear array

d = perpendicular distance traveled by the plane wave as it moves from one element to the next

α = a conical angle

t_l = time delay

t_m = measured time delay

c_{11} = angle of rotation

v = noise term

σ_t = standard deviation of the time delay noise

σ_α = standard deviation of the noise on α

k = time count

h = sampling period in seconds

ψ = observer's heading angle

∇ = gradient operator

$||x||$ = the norm of x

x^T = the transpose of x

PI1 = performance index for the nonlinear problem

PI2 = performance index for the linearized problem

g_i = "direction" of ith step (not unit magnitude)

a_i = length of the ith step

$x = (V_{xt}, V_{yt}, R_{xt}(0), R_{yt}(0), R_{zt}(0))^T$

$f_k(x)$ = the kth measurement residual

LIST OF SYMBOLS (Cont'd)

R_S = slant range

R_{11}, R_T = linear array coordinates

V_{xt}, V_{yt}, R_{xt}, R_{yt}, R_{zt} = target parameters

V_{xo}, V_{yo}, R_{xo}, R_{yo}, R_{zo} = observer parameters

V_x, V_y, R_x, R_y, R_z = target-observer parameters

K = total number of measurements

n = number of unknowns

a_m = the value of a_i which minimizes the quadratic polynomial

$A = (J \vdots f)$

UNDERWATER TARGET TRACKING USING MEASUREMENTS FROM A LINEAR ARRAY OF OMNIDIRECTIONAL ACOUSTIC ELEMENTS

1. INTRODUCTION

The tracking problem discussed herein is defined by a moving observer in an underwater environment which uses measurements from a trailing linear acoustic array to estimate the initial position and velocity of a target. The assumptions made in discussing this problem are as follows. The array is a multispot linear array of uniformly-spaced omnidirectional elements operating in a passive mode; it is considered to be an inflexible body which is rigidly connected to the observer's stern. The target-observer kinematics are described in a three-dimensional Cartesian coordinate system. Both the velocity and depth of the target are constant throughout the problem. The observer maneuvers to improve the quality of the estimate and these maneuvers are generated by impulses in acceleration and are restricted to a horizontal plane at a depth that may differ from the target's. The underwater environment is modeled as an infinite, homogeneous medium with the target being the only noise source in this medium. Finally, it is assumed that the linear array is operating in the far-field region, so that only plane-wave propagation is considered.

Various versions of this target tracking problem are produced by changing the type of sensor, altering the restrictions on the environment, and relaxing the restrictions on target motion and other assumptions. One well-studied version is the bearings-only case, where both the target and observer are operating at the same depth and a bow-mounted sonar measures the azimuthal angle between north and the line-of-sight to the target. In other words, by steering in a horizontal plane, this sonar isolates a line segment on which the target lies. In contrast with this, the linear array measures a conical angle between the array's longitudinal center line and the surface of a cone on which the target lies.

The measurement equation describing this conical angle is a nonlinear function of the target parameters: velocity and initial position. Consequently, an iterative least-squares estimator is used. Although the extended Kalman filter is also applicable, it was not tried because of its erratic behavior with the Cartesian model of bearings-only target motion analysis,[1] which has many of the same properties as the omni-directional linear array problem. Also, the additional experience with iterative techniques may prove useful in treating other problems where past measurements from secondary sensors must be incorporated into the current estimate. The measurement noise is assumed to be Gaussian even though the iterative least-squares estimator does not require any information about the noise structure.

The modified Gauss-Newton method[2] is the particular iterative technique used in this study. However, its unmodified version, the Gauss-Newton method, is also discussed and some results obtained with it are cited. The performance index, PI1, for this nonlinear least-squares problem is the sum of the squared measurement residuals, which is minimized by iteratively estimating the target parameters. The previous estimate defines a nominal target track which is used to formulate a linearized least-squares problem that is solved by a Householder transformation.[3] This solution updates the previous estimate, and the procedure repeats until the terminating criteria are satisfied. The modified Gauss-Newton method is used because of its elementary structure, and the Householder transformation is used because of its numerical accuracy.

The next section of this report develops the Cartesian signal model required by the estimation algorithm. This is followed by a section that describes the Gauss-Newton method, the modified Gauss-Newton method, and the Householder transformation. The Levenberg-Marquardt method[4] is also mentioned because of its success with singular or ill-conditioned

problems. The experimental results are given in the fourth section, and the conclusions and recommendations for further work are given in the last section.

Appendix A shows the Jocobian matrix required in the Gauss-Newton method, and appendix B shows a scheme for selecting the step size in the modified Gauss-Newton method. The number of computer operations (additions, multiplications, divisions, square roots, and trigonometric functions) required for one iteration of the modified Gauss-Newton method is given in appendix C. A listing of the computer code used to define the experiments is contained in appendix D.

2. SIGNAL MODEL

The signal model is defined by the measurement process and the target-observer kinematics, and both of these are described in a three-dimensional Cartesian coordinate system in this section. The complexity of this description is reduced because neither the target nor the observer changes depth. A further simplification results from the assumption on the observer's maneuvers; i.e., instantaneous changes in speed and heading are permitted. Consequently, the center lines of the observer and the linear array are always coincident. This gives a shorter convergence time when compared with the results from a signal model that uses a realistic description of the maneuvers.

Figure 1 shows the two-element model of the linear array used in this report. The distance between elements is b, and the distance traveled by the plane wave as it moves from the first to the second element is d. The direction to the target is given by the angle α, which should be defined at the midpoint between the two elements. This error is neglected, as is the offset between the positions of the array and the observer.

Inspection of figure 1 shows that

$$\cos \alpha = d/b = c\, t_1/b, \tag{2-1}$$

where c is the nominal speed of sound in sea water and t_1 is the wave front time delay between elements. The observer reconstructs the plane wave by combining the electrical signals from these two elements after first inserting the time delay, t_1, into the signal path of the second element. Further inspection of figure 1 shows that α and, consequently, t_1 are unchanged when the direction of propagation line is rotated

about the array's longitudinal center line. Hence, a time delay measurement defines the surface of a cone on which the target lies. This is shown in figure 2.

In this report t_1 rather than α is defined as the measured variable because $\cos \alpha$ is the required input signal for the estimation algorithm presented in the next section. Assuming that α is the measured variable, the noise level on t_1 is established from equation (2-1). The noisy version of this equation is

$$\cos(\alpha + v_\alpha) = c(t_1 + v_t)/b \tag{2-2}$$

where v_α and v_t are the noise terms. Expanding the cosine term and simplifying the result gives

$$v_t = v_\alpha (b/c) \sin \alpha \tag{2-3}$$

Assuming that $v_\alpha \sim N(0, \sigma_\alpha{}^2)$ and that at any point in time α is unknown but constant over an ensemble of experiments, then $v_t \sim N(0, \sigma_t{}^2)$ where

$$\sigma_t = b|\sin \alpha|\sigma_\alpha/c. \tag{2-4}$$

Hereafter,

$$\sigma_t = b\, \sigma_\alpha/c \tag{2-5}$$

is used. This approximation leads to a pessimistic result.

The noisy time delay is defined as t_m; i.e.,

$$t_m = t_1 + v \tag{2-6}$$

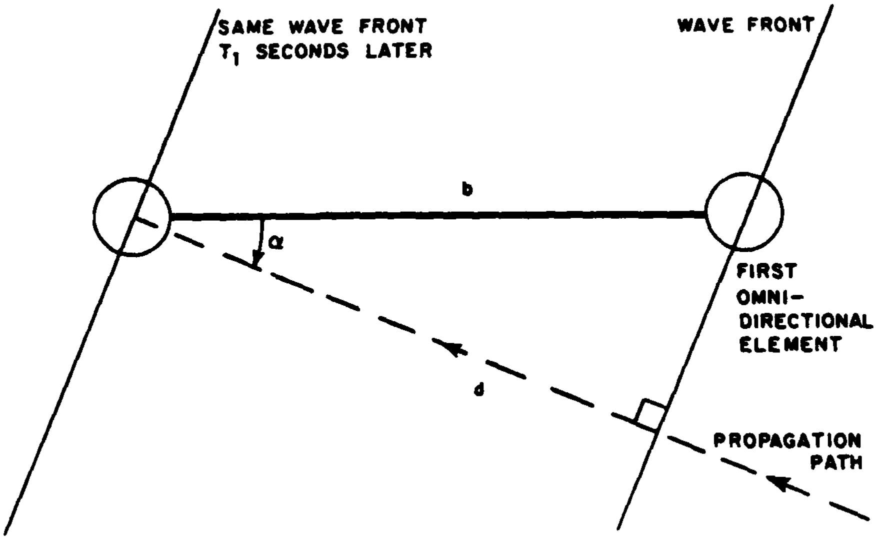


Figure 1. A Plane Wave Arriving at a Two-Element Omnidirectional Linear Array

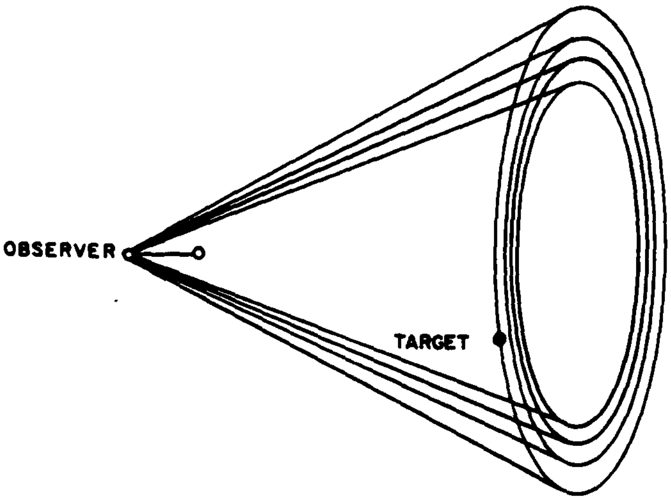


Figure 2. The Surfaces of Four Cones, Each for a Different Time Delay

where the t subscript on v has been dropped for brevity. Solving equation (2-1) for t_1 and substituting this result into equation (2-6) yields

$$t_m(k) = (b \cos \alpha(k))/c + v(k) \tag{2-7}$$

where k denotes the time t = kh at which the measurement is taken with a fixed sampling period h seconds. Equation (2-7) defines the measured time delay as a function of the conical angle α. To define it as a function of the target's position variables, a Cartesian coordinate system is introduced.

A (R_{11}, R_T, R_z) coordinate system fixed at the array with R_{11} along the center line, R_z in the depth direction, and R_T orthogonal to both R_{11} and R_z is shown in figure 3. Inspection of this figure shows that

$$\cos \alpha = R_{11}/R_s \tag{2-8}$$

where

$$R_s = (R_{11}{}^2 + R_T{}^2 + R_z{}^2)^{1/2} \tag{2-9}$$

is the slant range to the target. Substituting equation (2-8) into equation (2-7) gives

$$t_m\ (k) = b\ R_{11}\ (k)/c\ R_s(k) + v(k), \tag{2-10}$$

which shows the measurement as a function of the target's relative position variables. Also, from figure 3, it is easily shown that

$$R_{11}^2 - R_s^2 \cos^2 \alpha = 0 \qquad (2\text{-}11)$$

defines the surfaces of the cone on which the target lies.

Because the (R_{11}, R_T) plane rotates when the observer changes course, these variables must be transformed to a north-referenced coordinate system where the constant velocity target is defined. The new variables are R_y (which points north) and R_x (which is orthogonal to both R_y and R_z). The (R_x, R_y) and (R_{11}, R_T) planes are coincident but differ by a rotation, and R_z is the same in both coordinate systems. Both of these coordinate systems are fixed at the array.

This rotation is defined by the transformation shown in the next equation.

$$\begin{bmatrix} R_X \\ R_Y \end{bmatrix} = \begin{bmatrix} \cos c_{11} & \sin c_{11} \\ -\sin c_{11} & \cos c_{11} \end{bmatrix} \begin{bmatrix} R_T \\ R_{11} \end{bmatrix}, \qquad (2\text{-}12)$$

where c_{11} is the angle betwen the R_Y and the R_{11} axes, measured positive clockwise from the north reference. Because this is an orthogonal transformation, R_s is also given by

$$R_s = (R_x^2 + R_Y^2 + R_z^2)^{1/2} . \qquad (2\text{-}13)$$

Figure 4 shows these two Cartesian coordinate systems, the array, and the observer's heading angle.

This transformation is expressed in terms of the observer's heading angle, ψ, by substituting

$$c_{11} = \psi - 180^\circ \qquad (2\text{-}14)$$

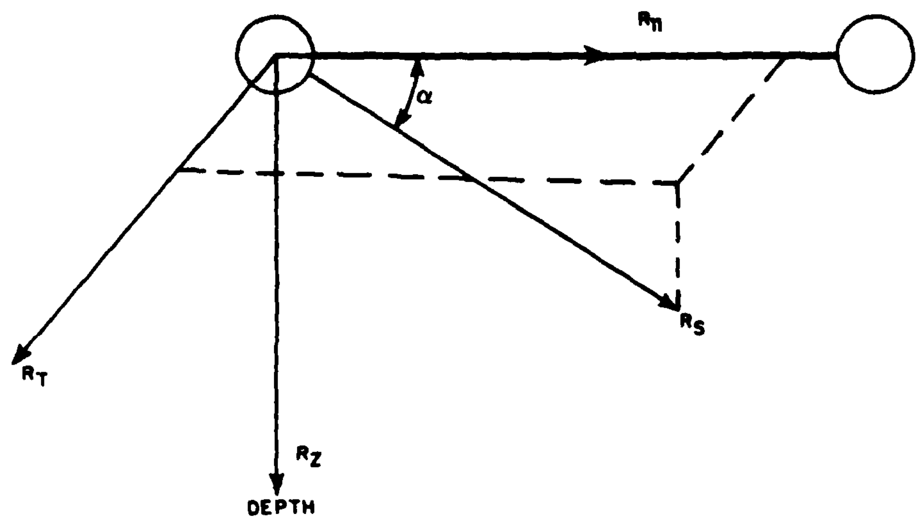


Figure 3. The (R_{11}, R_T, R_Z) Coordinates System Fixed at the Array

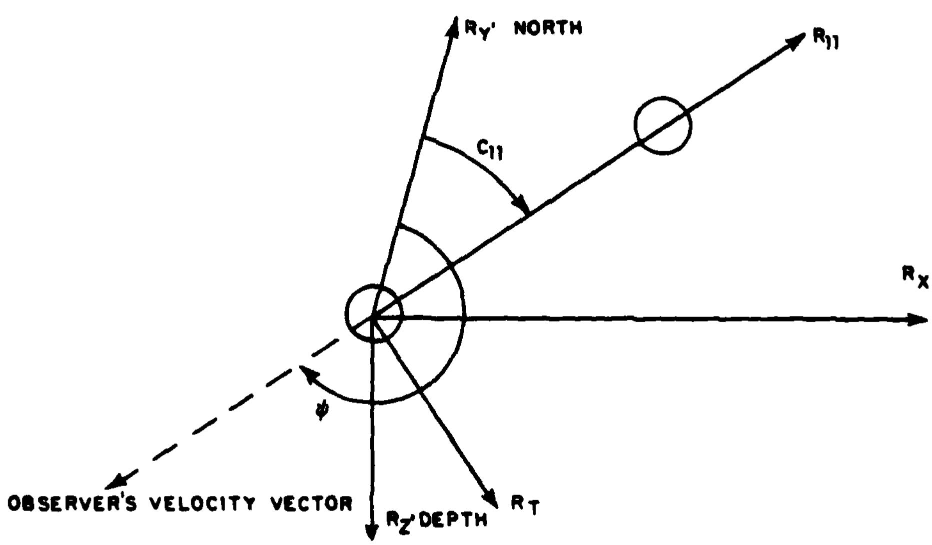


Figure 4. Two Coordinate Systems Fixed at the Array

into equation (2-12). After first multiplying equation (2-12) by the inverse transformation, this gives

$$\begin{bmatrix} R_T \\ R_{11} \end{bmatrix} = \begin{bmatrix} -\cos\psi & \sin\psi \\ -\sin\psi & -\cos\psi \end{bmatrix} \begin{bmatrix} R_X \\ R_Y \end{bmatrix} . \qquad (2\text{-}15)$$

Using the last equation in the matrix equation (2-15) to eliminate R_{11} from equation (2-10) gives

$$t_m(k) = -b(R_x(k)\sin\psi(k) + R_Y(k)\cos\psi(k))/c\ R_s(k) + v(k) \qquad (2\text{-}16)$$

where R_s is given by equation (2-13). Equation (2-16) shows the measured variable as a function of relative position variables which are defined in a north-referenced system.

Inspection of equation (2-16) shows that the sign on R_z cannot be resolved by this measurement. This was demonstrated by simulation, although this elementary experiment is not included in this report.

For a constant velocity target, these relative position variables are defined by

$$R_x(k) = V_{xt}kh + R_{xt}(0) - R_{xo}(k) \qquad (2\text{-}17a)$$

$$R_Y(k) = V_{yt}kh + R_{yt}(0) - R_{yo}(k) \qquad (2\text{-}17b)$$

$$R_z(k) = R_{zt}(0) \qquad (2\text{-}17c)$$

where V_{xt}, V_{yt}, $R_{xt}(0)$, $R_{yt}(0)$, and $R_{zt}(0)$ define the target paramters in another north-referenced coordinate system which is fixed in

space and which at $t = 0$ is coincident with the north-referenced system fixed to the observer. Also, $R_{xo}(k)$ and $R_{yo}(k)$ define the observer's position in this stationary coordinate system.

Equation (2-16) defines the measurement process and equation (2-17) defines the target-observer kinematics. Together, these equations define the signal model required by the estimation algorithm presented in the next section.

3. ITERATIVE LEAST-SQUARES TECHNIQUE

Because the measurement equation (2-16) is nonlinear, an interative least-squares algorithm which batch processes a set of measurements is selected to estimate the target parameters. The extended Kalman filter could be used for this problem, but it was not tried because its performance with a Cartesian signal model of the bearings-only problem is suspect.

In this least-squares problem the modified Gauss-Newton method is used to iteratively minimize the sum of the squared measurement residuals. At each iteration the Gauss-Newton equations are solved by a Householder transformation, and this yields an increment or step which is added to the previous estimate. The modified Gauss-Newton method first multiplies this step by a positive scalar, which is selected at each iteration. The Levenburg-Marquardt method is also mentioned in this section; however, it did not demonstrate any advantage in the experiments, so its use is not recommended at this time. These experiments are not contained in this report.

The sum of the squared measurement residuals which defines the performance index PI1 is given by

$$PI1(x) = ||f(x)||^2 = f^T(x)f(x) = \sum_{k=0}^{K-1} f_k^2(x) \tag{3-1}$$

where $K > 5$, $x^T = (V_{xt}, V_{yt}, R_{xt}(0), R_{yt}(0), R_{zt}(0))$,

$f^T(x) = (f_0(x), f_1(x), \ldots, f_{K-1}(x))$, and

$$f_k(x) = (R_x(k)\sin\psi(k) + R_y(k)\cos\psi(k)) + c\,R_s(k)t_m(k)/b \tag{3-2}$$

In this section x represents the estimate of the target parameters. Equation (3-2) defines the kth measurement residual. With K finite, it is expected that the true target parameters can minimize (3-1) only when the measurements are noiseless.

A general iterative procedure for minimizing PI1 is given by

$$x_{i+1} = x_i + g_i \tag{3-3}$$

where x_{i+1} is the (i+1)th estimate, x_i is the ith estimate, and g_i is the ith step. In theory, the initial estimate x_0 is usually selected as zero. Unfortunately, this did not work well, so the first measurement is used to aid in the selection of x_0. The procedure terminates in a finite number of iterations when no further significant improvement in PI1 occurs and/or when $||x_{i+1} - x_i||_{max}$ is less than some threshold. A requirement for convergence of this procedure using the modified Gauss-Newton method is given in reference 5. Although convergence proofs are beneficial, they do not guarantee the performance of the algorithm. Even though convergence is assured, double precision arithmetic and/or an excessive number of iterations may be needed.

The linearized least-squares problem which is solved at each iteration is derived from the following Taylor series expansion of PI1:

$$PI1(x_{i+1}) = PI1(x_i+g_i) = PI1(x_i) + \nabla PI1^T(x_i)g_i + \text{higher-order terms} \tag{3-4}$$

where ∇ is the gradient operator. If x_{i+1} yields a local minimum, then $\nabla PI1(x_{i+1}) = 0$, and the gradient of equation (3-4) gives

$$0 = \nabla PI1(x_i) + \nabla^2 PI1(x_i)g_i \tag{3-5}$$

where the higher-order terms have been neglected. The step g_i to the next estimate is given by the solution to the matrix equation

$$\nabla^2 PI1(x_i) g_i = \nabla PI1(x_i) \tag{3-6}$$

where

$$\nabla PI1(x) = \partial f^T(x)\, f(x)/\partial x = 2\, J^T(x)\, f(x) \tag{3-7}$$

and

$$\nabla^2 PI1(x) = 2\, J^T(x)\, J(x) + 2 \sum_{k=0}^{K-1} (\nabla^2 f_k(x))\, f_k(x). \tag{3-8}$$

J(x), the K by 5 Jacobian matrix of f(x), is given by

$$J(x) = \partial f(x)/\partial x = [\partial f_k/\partial x_i], \tag{3-9}$$

which is given in appendix A. This appendix shows that the last column of J(x) is zero if the estimate of the target's depth is zero. This singularity did not present any difficulties for the experiments reported here; it will be discussed in detail in a future report.

The computer burden is reduced if the last term in equation (3-8) is replaced by an approximation which does not require any second derivatives. Because this term depends upon the residuals, the approximation usually introduces negligible error near the solution. Various iterative techniques result from different approximations. Using equation (3-8) as stated gives Newton's method, while dropping the last term gives the Gauss-Newton method. In addition, setting $\nabla^2 PI1(x) = I$, an identity matrix, gives a steepest decent step $g_i = -\nabla PI1(x_i)$. The

Levenberg-Marquardt method provides a mix of these latter two techniques. Matrix-updating methods, such as the Davidon-Flectcher-Powell method, approximate the Hessian matrix, $\nabla^2 PI$, by a successive addition of low-rank matrices.

The convergence rate for Newton's method is second order, while reference 5 states that at most the convergence rate for the Gauss-Newton method is linear unless $PI1(x_{min}) = 0$, in which case it is ultimately second order.

Dropping the last term in equation (3-8) gives

$$J^T(x_i)\ J(x_i)\ g_i = -J^T(x_i)\ f(x_i), \tag{3-10}$$

which defines the normal equations for the Gauss-Newton linearization. Solving these equations for g_i minimizes a linearized performance index given by

$$PI2(g_i) = \|J(x_i)\ g_i - (-f(x_i))\|^2. \tag{3-11}$$

These normal equations (3-10) are solved by applying a Householder transformation to the squared error shown in equation (3-11). This numerically more accurate approach[6] is developed by introducing $T^T T$ into equation (3-11) as follows:

$$PI2(g_i) = (J\ (x_i)\ g_i + f(x_i))^T\ T^T T(J(x_i)\ g_i + f(x_i))$$

$$PI2(g_i) = (T\ J(x_i)\ g_i + T\ f(x_i))^T (T\ J(x_i)\ g_i + T\ f(x_i))$$

$$PI2(g_i) = \|T\ J\ (x_i)\ g_i + T\ f\ (x_i)\ \|^2 \tag{3-12}$$

where T is the Householder transformation. Because this is an orthogonal transformation, $T^T T = I$ and inserting this into equation (3-11) leaves PI2 unchanged. In addition, T is selected such that

$$T\,J = \left(\frac{R}{0}\right) \tag{3-13}$$

where R is a 5 by 5 upper triangular matrix and 0 is a (K-5) by 5 zero matrix. This gives

$$PI2 = \left\|\left(\frac{R}{0}\right) g_i + \left(\frac{\bar{f}}{e}\right)\right\|^2 \tag{3-14}$$

where

$$T\,f = \left(\frac{\bar{f}}{e}\right). \tag{3-15}$$

The g_i which minimizes PI2 is given by

$$g_i = -R^{-1}\,\bar{f} \tag{3-16}$$

and

$$\min PI2 = \|e\|^2. \tag{3-17}$$

The details associated with defining the elements of this transformation T are given in references 3, 6, and 7.

In the vicinity of a local minimum at x_{i+1},

$$f(x_{i+1}) = f(x_i + g_i) = f(x_i) + J(x_i)\,g_i \tag{3-18}$$

and, consequently, (3-11) becomes

$$PI2(g_i) = ||f(x_{i+1})||^2 = PI1(x_{i+1}). \tag{3-19}$$

Equation (3-19) is not a sufficient condition for a local minimum, but it is convenient to use it as the termination criterion for the experiments.

The modified Gauss-Newton method is given by

$$X_{i+1} = x_i + a_i\, g_i \tag{3-20}$$

where the positive scalar a_i is selected such that

$$PI1(x_{i+1}) < PI1(x_i). \tag{3-21}$$

In this method it is said that g_i gives the direction of the step while a_i gives its length.

Once g_i is found, by the Gauss-Newton method, $PI1(x_{i+1}) = PI1(x_i + a_i\, g_i)$ is only a function of the scalar a_i; therefore, this function can be minimized by the selection of a_i. There are several schemes for selecting a_i. The one used in this report finds the minimum of a quadratic polynominal which fits the (a_i, PI2) data in the g_i direction. This is generated by computing PI2 for three different values of a_i. An analytical expression for the minimum in terms of these three data points is derived in appendix B.

Inspection of equation (3-10) shows that a non-singular J^TJ matrix is required by the Gauss-Newton method. Ill-conditioning of this matrix is also a concern. The Levenberg-Marquardt method is useful in these situations, and it also provides a way to incorporate a priori information into the problem.

The Levenberg-Marquardt method is defined by equation (3-5) but with

$$\nabla^2 PII(x_i) = 2\, J^T(x_i)\, J(x_i) + 2\, \lambda_i\, D \qquad (3\text{-}22)$$

where λ_i is a positive scalar and D is a positive definite diagonal matrix. By comparing equation (3-22) with equation (3-8) it is seen that $2\, \lambda_i\, D$ may be viewed as an approximation to the term that is omitted by the Gauss-Newton method. The details of the selection procedure for both λ_i and D are given in reference 4, but inspection of equation (3-22) shows that $\lambda_i = 0$ yields the Gauss-Newton method while $\lambda_i \to \infty$ with $D = I$ yields steepest descent.

In the experiments reported in the next section, no difficulties with the $J^T J$ matrix occurred, so the Levenberg-Marquardt method did not prove useful. These experimental results are not shown in this report. The Levenberg-Marquardt method is discussed because the omnidirectional linear array problem is singular prior to an observer maneuver, and estimates during this time interval may be required.

4. EXPERIMENTAL RESULTS

The results from a digital computer simulation of three different sets of experiments are presented in this section. The different sets are produced by changing the target's initial position. Each of these sets is composed of three members, which are produced by changing the standard deviation of the measurement noise. The target parameters for these three sets of experiments are given in table 1.

Table 1. True Target Parameters with Speed in Meters/Second and Range in Meters

Experiment Set	V_{xt}	V_{yt}	$R_{xt}(0)$	$R_{yt}(0)$	$R_{zt}(0)$
No. 1	-5	10	10,000	10,000	1000
No. 2	-5	10	40,000	40,000	0
No. 3	-5	10	10,000	10,000	10,000

Experiment set No. 1 is a short-to-mid range situation while No. 2 is a mid-to-long range one. Experiment set No. 3 may be viewed as a single-bottom-bounce problem where the actual target depth is located by knowing the bottom depth. This is shown in figure 5.

The standard deviation for the white Gaussian measurement noise is established from equation (2-5) with c = 1500 meters/second and b = 3/8 meter (which is a quarter wave-length at a 1-kHz frequency). The three

standard deviations used in each of the sets shown in table 1 are $\sigma_t = 0.0$ seconds, $\sigma_t = 0.05 \times 10^{-5}$ seconds, and $\sigma_t = 0.25 \times 10^{-5}$ seconds. The first of these is the noiseless case while the last corresponds to a $\sigma_\alpha = 0.01$ radians (approximately 0.5 degrees). The remaining σ_t represents a low-noise case.

In all experiments the observer starts from the origin at $t = 0$ seconds with a speed of 10 meters/second and a heading of 0 degrees. Just prior to the measurement at 240 seconds the observer instantaneously changes heading to 90 degrees and just prior to 480 seconds changes back to 0 degrees. Each of these headings defines a leg of the problem, and each leg consists of 240 measurements. The first measurement occurs at 0 seconds while the last occurs at 719 seconds. With the exception of the target's initial position, the target and observer tracks are the same for all experiments. Figure 6 shows these tracks projected onto the (R_y, R_x) plane.

The time-delay measurements are taken at a 1-second data rate; they are then compressed by a measurement preprocessor, which also reduces the noise. For example, this preprocessor produces a "measurement" at 9.5 seconds by first adding together the measurements from 0 to 19 seconds and then dividing this sum by 20. These preprocessed measurements are then fed to the estimation algorithm. Hereafter, the indices of all the variables denote the preprocessed measurement count; e.g., $t_m(0)$ is the first of these measurements and it is associated with a real time of 9.5 seconds while $t_m(1)$ is the second, and real time is 29.5 seconds. The error introduced by measurement preprocessing is not modeled in these experiments. This error is zero only if the true time delay is constant for each of the 20-second intervals. Finally, the time-delay preprocessor averages $\cos \alpha$, not α itself.

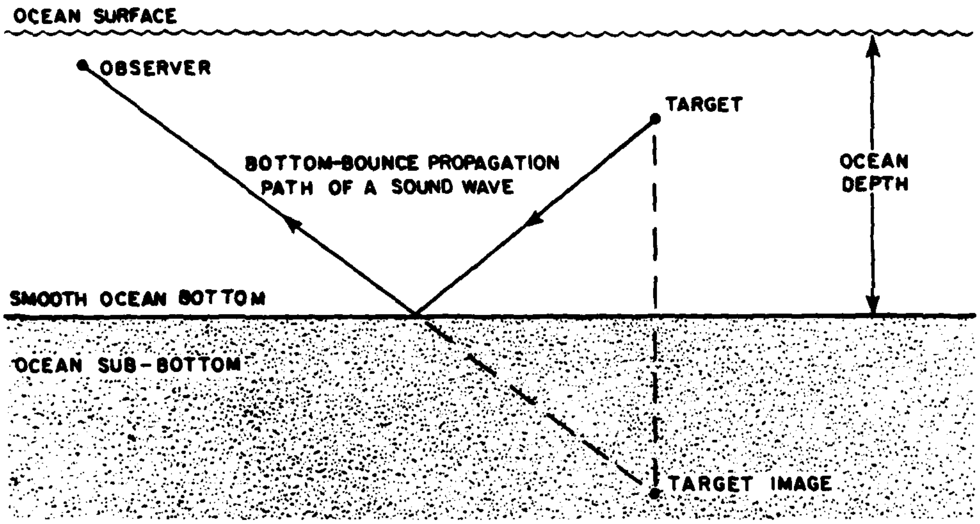


Figure 5. A Single Bottom-Bounce Propagation Path

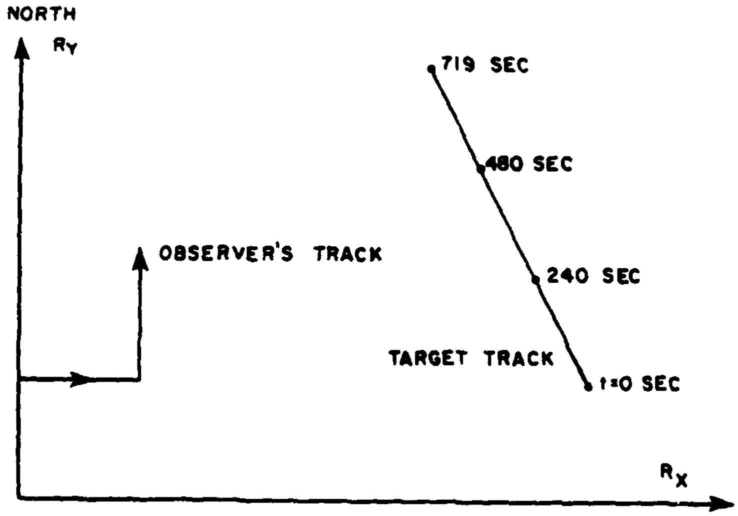


Figure 6. Target and Observer Tracks in the (R_y, R_x) Plane

To start the iterative least-squares algorithm an initial estimate of the target parameters is required. This is specified by using the first measurement together with $V_{xt} = 0$, $V_{yt} = 0$, $R_{zt}(0) = 3{,}000$ meters, and $R_s(0) = 20{,}000$ meters; this gives,

$$R_{11}(0) = c\, R_s(0)\, t_m(0)/b \tag{4-1}$$

and

$$R_T(0) = \pm\, (R_s{}^2(0) - R_{zt}{}^2(0) - R_{11}{}^2(0))^{1/2} \tag{4-2}$$

where the sign on $R_T(0)$ determines the initial side of the cone's surface on which the target is assumed to by lying. Both signs are tried and the one which yields the smaller PI1 is selected as the initial point. Although this does require that the estimate of the target-observer track be generated twice, it does not require a Householder transformation.

Another initialization scheme is to select $R_T(0) = 0$ and

$$R_{zt}(0) = \pm\, (R_s{}^2(0) - R_{11}(0))^{1/2}, \tag{4-3}$$

where either sign may be used without penalty. This scheme may also be used whenever the quantity under the square-root operator in equation (4-2) is negative. A future report will investigate the use of equation (4-3) and $R_T(0) = 0$ as the primary initialization procedure.

In these experiments the iterative procedure is terminated when $PI1(x_{i+1})$ is within $\pm$ 10 percent of $PI2(g_1)$. This criterion is selected as a convenience for these initial experiments. It is recommended that a similar bound be placed upon the component of x_{i+1}, which undergoes the largest percent change. Monitoring the percent change in PI1 is also recommended.

The first experiments use the modified Gauss-Newton method. The initialization uses the sign on $R_T(0)$ which produces the smaller PI1, so the estimates associated with generating the additional target-observer track are not shown in the results. However, some additional experiments do show the effects of using the "wrong" sign on $R_T(0)$. This is followed by some results for both a one-legged and a two-legged problem. Finally, results obtained with the Gauss-Newton method are mentioned.

The results from the first experiment set are shown in tables 2a, 2b, and 2c; from the second in tables 3a, 3b, and 3c; and from the third in tables 4a, 4b, and 4c. In all of these tables the iteration number defines the number of calls to the Householder transformation subroutine, and the last row shows the PI1 which occurs when the true target parameters are used. As expected, this PI1 is larger than the one reached by the algorithm in all cases.

Inspection of all these tables shows that the algorithm converges by the fourth iteration, except for the noiseless cases. In these cases it is seen that $R_{zt}(0)$ is the only target parameter which experiences more than a 10 percent decrease in moving from the third to the fourth iteration, and this only occurs in tables 2a and 3a. Perhaps, after the fourth iteration a 1-dimensional search or a less stringent performance bound should be used for $R_{zt}(0)$. Regardless, further investigation or discussion of these noiseless cases is not warranted.

Inspection of tables 2a, 2b, and 2c shows that the magnitude of the error in the final estimate increases, as expected, when the variance of the noise is increased. This is also shown in the other sets of tables. A Monte Carlo simulation or an analytical study could quantify this observation.

Table 2a. Results from the Modified Gauss-Newton Method with σ_t = 0.0 Seconds (First Member of Set No. 1)

Iteration Number	V_{xt}	V_{yt}	$R_{xt}(0)$	$R_{yt}(0)$	$R_{zt}(0)$	PI1	PI2
0	0	0	13,800	14,161	3000	3.79×10^8	4.70×10^5
1	2.75	10.25	6179	6993	3087	1.89×10^7	8463
2	-2.80	12.99	8742	8563	1701	5.87×10^5	46.47
3	-5.39	9.64	10,230	10,233	1127	3561	0.024
4	-4.99	10.01	9997	9997	1006	4.58	0.057
5	-5.00	10.00	10,002	10,002	1002	0.059	----
True Parameters	-5.00	10.00	10,000	10,000	1000	0.57	----

Table 2b. Results from the Modified Gauss-Newton Method with $\sigma_t = 0.05 \times 10^{-5}$ Seconds (Second Member of Set No. 1)

Iteration Number	V_{xt}	V_{yt}	$R_{xt}(0)$	$R_{yt}(0)$	$R_{zt}(0)$	PI1	PI2
0	0	0	13,827	14,136	3000	3.83×10^8	4.75×10^5
1	2.70	10.25	6178	6993	3109	1.86×10^7	8096
2	-3.05	12.89	8899	8699	1734	5.88×10^5	763
3	-5.56	9.53	10,328	10,326	1165	4014	734
4	-5.18	9.89	10,106	10,100	1054	737	----
True Parameters	-5.00	10.00	10,000	10,000	1000	1029	----

Table 2c. Results from the Modified Gauss-Newton Method with σ_t = 0.25 X 10^{-5} Seconds (Third Member of Set No. 1)

Iteration Number	V_{xt}	V_{yt}	$R_{xt}(0)$	$R_{yt}(0)$	$R_{zt}(0)$	PI1	PI2
0	0	0	13,801	14,161	3000	3.79×10^8	5.02×10^5
1	2.98	10.00	6285	7058	2939	2.38×10^7	42,580
2	-1.22	13.52	7637	7572	1526	6.52×10^5	29,736
3	-4.24	10.47	9494	9451	858	37,858	29,648
4	-3.75	10.84	9191	9166	613	29,739	----
True Parameters	-5.00	10.00	10,000	10,000	1000	37,793	----

Table 3a. Results from the Modified Gauss-Newton Method with σ_t = 0.0 Seconds (First Member of Set No. 2)

Iteration Number	V_{xt}	V_{yt}	$R_{xt}(0)$	$R_{yt}(0)$	$R_{zt}(0)$	PI1	PI2
0	0	0	13,812	14,150	3000	5.51×10^7	22,182
1	17.56	30.92	26,420	26,916	4659	3.64×10^6	2041
2	-5.74	10.06	41,391	41,203	2628	1.37×10^5	0.148
3	-4.85	9.99	39,729	39,766	1013	3813	0.148
4	-5.02	9.99	40,037	40,032	454	98	0.140
5	-5.00	10.00	39,995	39,996	178	3	0.142
6	-5.00	10.00	40,003	40,003	101	0.173	0.137
7	-5.00	10.00	40,004	40,004	89	0.163	0.134
8	-5.00	10.00	40,004	40,004	87	0.134	-----
True Parameters	-5.00	10.00	40,000	40,000	0.0	missing	

Table 3b. Results from the Modified Gauss-Newton Method with $\sigma_t = 0.05 \times 10^{-5}$ Seconds (Second Member of Set No. 2)

Iteration Number	V_{xt}	V_{yt}	$R_{xt}(0)$	$R_{yt}(0)$	$R_{zt}(0)$	PI1	PI2
0	0	0	13,806	14,156	3000	5.46×10^7	30,848
1	17.60	30.94	26,416	26,921	4668	3.62×10^6	25,604
2	-5.70	9.92	41,177	41,016	2663	1.54×10^5	23,343
3	-4.83	9.92	39,558	39,616	1131	26,524	23,343
4	-4.98	9.91	39,820	39,844	754	23,378	----
True Parameters	-5.00	10.00	40,000	40,000	0.0	24,205	----

Table 3c. Results from the Modified Gauss-Newton Method with $\sigma_t = 0.25 \times 10^{-5}$ Seconds (Third Member of Set No. 2)

Iteration Number	V_{xt}	V_{yt}	$R_{xt}(0)$	$R_{yt}(0)$	$R_{zt}(0)$	PI1	PI2
0	0	0	13,757	14,204	3000	5.04×10^7	2.03×10^5
1	19.54	32.84	25,919	26,460	4549	3.83×10^6	3.31×10^5
2	-7.51	7.22	41,133	41,199	2844	4.00×10^5	3.04×10^5
3	-6.94	7.14	40,120	40,346	1880	3.06×10^5	----
True Parameters	-5.00	10.00	40,000	40,000	0.0	3.77×10^5	----

Table 4a. Results from the Modified Gauss-Newton Method with σ_t = 0.0 Seconds (First Member of Set No. 3)

Iteration Number	V_{xt}	V_{yt}	$R_{xt}(0)$	$R_{yt}(0)$	$R_{zt}(0)$	PI1	PI2
0	0	0	16,039	11,565	3000	8.69×10^8	3.91×10^5
1	-1.66	1.58	13,726	11,045	10,376	4.63×10^8	1.49×10^5
2	-6.58	6.73	9975	10,586	10,905	1.34×10^7	1163
3	-5.16	10.11	10,196	10,112	10,059	47,132	0.0066
4	-5.00	10.00	10,000	10,000	10,001	0.522	0.0073
5	-5.00	10.00	9999	10,000	10,000	0.0061	0.0061
6	-5.00	10.00	9999	10,000	10,000	0.0060	----
True Parameters	-5.00	10.00	10,000	10,000	10,000	0.261	----

Table 4b. Results from the Modified Gauss-Newton Method with $\sigma_t = 0.05 \times 10^{-5}$ Seconds (Second Member of Set No. 3)

Iteration Number	V_{xt}	V_{yt}	$R_{xt}(0)$	$R_{yt}(0)$	$R_{zt}(0)$	PI1	PI2
0	0	0	16,036	11,569	3000	8.68×10^8	3.80×10^5
1	-1.67	1.58	13,730	11,049	10,375	4.63×10^8	1.42×10^5
2	-6.55	6.75	9950	10,558	10,886	1.39×10^7	2579
3	-5.00	10.12	10,083	10,017	9980	49,698	1815
4	-4.86	10.01	9900	9915	9931	1816	----
True Parameters	-5.00	10.00	10,000	10,000	10,000	2003	----

Table 4c. Results from the Modified Gauss-Newton Method with σ_t = 0.25 X 10^{-5} Seconds (Third Member of Set No. 3)

Iteration Number	V_{xt}	V_{yt}	$R_{xt}(0)$	$R_{yt}(0)$	$R_{zt}(0)$	PI1	PI2
0	0	0	16,065	11,529	3000	8.77×10^{8}	4.92×10^{5}
1	-1.70	1.54	13,718	10,995	10,364	4.65×10^{8}	2.28×10^{5}
2	-6.87	6.54	10,059	10,609	10,890	1.44×10^{7}	55,891
3	-5.67	9.97	10,440	10,247	10,152	1.01×10^{5}	53,939
4	-5.46	9.86	10,213	10,112	10,072	53,937	----
True Parameters	-5.00	10.00	10,000	10,000	10,000	60,327	----

Comparing tables 2c, 3c, and 4c shows that the final value of PI1 increases with increasing slant range. The results shown in these tables demonstrate that the modified Gauss-Newton method converges to a reasonable estimate of the target parameters in approximately four iterations.

The high noise cases for these three experiment sets were repeated, but they were initialized on the "wrong" side of the cone; i.e., the side which yields the larger PI1. The one from experiment set no. 3 displayed results similar to those reported in table 4c while the remaining two high noise cases performed poorly. After five iterations the one from experiment set no. 2 was stopped because convergence to a reasonable estimate was judged unlikely. After seven iterations the one from experiment set no. 1 was also stopped; however, convergence in one more iteration was expected but, unfortunately, with the wrong sign on the $R_{zt}(0)$ estimate.

To investigate the effects of fewer legs, the third experiment set shown in table 1 was repeated, first with two legs (0 to 479 seconds) and then with one (0 to 239 seconds). Table 5 shows the final results of the two-legged problem.

Table 5. Estimates from the First Two Legs of Experiment Set No. 3

Number of Iterations	σt	V_{xt}	V_{yt}	$R_{xt}(0)$	$R_{yt}(0)$	$R_{zt}(0)$	PI1
5	0.0	-5.0	10.0	10,000	10,000	10,000	0.0033
4	0.05×10^{-5}	-6.2	9.50	10,820	10,726	10,629	965.0
7	0.25×10^{-5}	10.0	2.55	-2495	1773	-90	852.0

This table shows that the noiseless and the low noise cases converged to reasonable estimates while the high noise case converged to a worthless result. This worthless result may be caused by fewer legs or by fewer measurements; additional experiments could resolve this point. As expected, the one-legged problem generated diverging results because the J^TJ matrix is singular. The Levenberg-Marquardt method could be used if an estimate on this leg is required.

Finally, the Gauss-Newton method was used to treat the three high noise cases. The results obtained were essentially the same as those shown in tables 2c, 3c, and 4c; consequently, the additional computations required to select a step size are not justified by these experiments. However, these experiments do not constitute an exhaustive set of all possible target-observer situations, and the modified Gauss-Newton method may yet prove useful.

5. CONCLUSIONS AND FURTHER WORK

The main conclusions are:

1. There is a sign ambiguity on the estimate of the target's depth.

2. The problem is singular on the first leg.

3. Both the Gauss-Newton method and the modified Gauss-Newton method converge in four iterations for these experiments.

4. The Jacobian matrix is singular if the estimate of the target's depth is zero.

5. Initialization of the iterative estimator affects convergence.

The sign on the target's depth, $R_{zt}(0)$, cannot be resolved by an observer operating at a constant depth. This is seen from equation (2-16), and it was also demonstrated experimentally although these results are not shown.

The target tracking problem with a linear array of omnidirectional elements is singular on the first leg. This was shown in the experiments. An analytic treatment of this point is not a high priority item, but is recommended since it would lend additional insight into this tracking problem.

Both the Gauss-Newton and the modified Gauss-Newton methods demonstrate convergence in approximately four iterations. Additional

experiments are needed to investigate convergence and, in those situations where convergence occurs, to determine if convergence in four iterations is typical.

Inspection of appendix A shows that the last column of the Jacobian matrix is zero if the estimate of the target's depth is zero; i.e., the target is estimated to be at the same depth as the observer. This singularity will be investigated in a subsequent report.

Initialization of the estimation algorithm can reduce the number of iterations needed for convergence. Further experiments on this item are needed.

To evaluate the Gauss-Newton method, more experiments with different target-observer tracks and with different noise variances are needed. A Monte Carlo simulation is also needed.

Different PI1 functions should be tried, with a view towards finding one that reduces the bias in the final estimate. Also, a probabilistic description of this problem might be more useful than the least-squares approach used here.

Experiments which expand the iterative least-squares estimator to inc.uded measurements from additional sensors would be useful.

Finally, results from an extended Kalman filter which explores the effects of different signal models should be documented.

6. REFERENCES

1. V. J. Aidala, "Kalman Filter Behavior in Bearings-Only Tracking Applications," IEEE Trans. Aerosp. Electron. Syst., Vol. AES-15, pp. 29-39, January 1979.

2. G. Dahlquist and A. Bjorck, Numerical Methods, Prentice-Hall, 1974.

3. G. J. Bierman, Factorization Methods for Discrete Sequential Estimation, Academic Press, 1977.

4. Y. Bard, Nonlinear Parameter Estimation, Academic Press, 1974.

5. S. A. Lill, (L.C.W. Dixon, ed.), "A Survey of Methods for Minimizing Sums of Squares of Nonlinear Functions," Optimization in Action, Academic Press, 1976.

6. C. L. Lawson and R. I. Hanson, Solving Least Squares Problems, Prentice-Hall, 1974.

7. M. F. Gouveia and D. J. Murphy, "Bearings-Only Target Tracking With a Square-Root Filter", Southeastern Massachusetts University Technical Report, June 1, 1980.

APPENDIX A. THE JACOBIAN OF f(x)

The Jacobian of f(x), which is used in the performance index of equation (3-1), is shown below. This f(x) is defined by equation (3-2).

The Jacobian is defined by

$$J = (\partial f_k/\partial x_j)$$

for $j = 1,2,3,4$, and 5; and $k = 0,1,2,\ldots,35$.

The entries in the kth row of J are given by:

$$\partial f_k/\partial x_1 = \partial f_k/\partial V_{xt} = kh\ \partial\ f_k/\partial R_{xt}(0)$$

$$\partial f_k/\partial x_2 = \partial f_k/\partial V_{yt} = kh\ \partial\ f_k/\partial R_{yt}(0)$$

$$\partial f_k/\partial x_3 = \partial f_k/\partial R_{xt}(0) = -\sin\psi\ (k) - ct_m(k)\ R_x(k)/b\ R_s(k)$$

$$\partial f_k/\partial x_4 = \partial f_k/\partial R_{yt}(0) = -\cos\psi\ (k) - ct_m(k)\ R_y(k)/b\ R_s(k)$$

$$\partial f_k/\partial x_5 = \partial f_r/\partial R_{zt}(0) = -ct_m(k)\ R_z(0)/b\ R_s(k).$$

APPENDIX B. STEP SIZE FOR THE MODIFIED GAUSS-NEWTON METHOD

In this appendix a scheme is given for selecting the step size a_i in the modified Gauss-Newton iterative formula.

$$x_{i+1} = x_i + a_i\, g_i \quad . \tag{B-1}$$

Actually, because it is not normalized, the direction g_i also contributes to the size of the step. It is convenient to redefine equation (B-1) as

$$x_{i+1} = x_i + a_j\, g_i \tag{B-2}$$

where a_j denotes the jth value of the step size at the ith iteration.

Once g_i is found from the Gauss-Newton equations, PI1 is a function only of a_j; i.e.,

$$PI1(a_j) = PI1(x_i + a_j\, g_i) \tag{B-3}$$

and this is minimized by a judicious selection of a_j. In this report, a_j is defined by the minimum of a quadratic polynomial which passes through three (a_j, PI1(a_j)) data points. For equally spaced values of a_j, the minimum of this quadratic is given by

$$a_m = \frac{(a_2+a_3)\ PI1(a_1) - 2(a_1+a_3)PI1(a_2) + (a_2+a_1)\ PI1(a_3)}{2\ PI1(a_1) - 4\ PI1(a_2) + 2\ PI1(a_3)} \tag{B-4}$$

where $\quad a_3 > a_2 > a_1 \geq 0$.

The first of these data points is readily available, namely, (a_1 = 0, PI1(0) = PI1(x_i)); and if

$$PI1(1) < PI1(0), \tag{B-5}$$

$a_2 = 1$ gives the second data point and $a_3 = 2a_2$ gives the third. However, if equation (B-5) is not satisfied the length of the interval is reduced by selecting $a_2 = 1/2$ and $a_3 = 2a_2 = 1$, provided

$$PI1(1/2) < PI1(0). \tag{B-6}$$

If this is unsuccessful, the next selection is $a_2 = 1/4$ and $a_3 = 2a_2 = 1/2$, and subsequent selections are given by repeatedly dividing a_2 by 2. This continues until $PI1(a_2) < PI1(a_1)$ or a threshold which causes termination of the estimation algorithm is crossed. After a_m is found, then $PI1(a_m)$, $PI1(a_2)$, and $PI1(a_3)$ are compared to determine which of these is the smallest. This is necessary because the quadratic polynomial may not always provide a good fit to the PI1 function and $PI1(a_2)$ or $PI1(a_3)$ may be smaller than $PI1(a_m)$. A diagram of this scheme is shown in figure B-1.

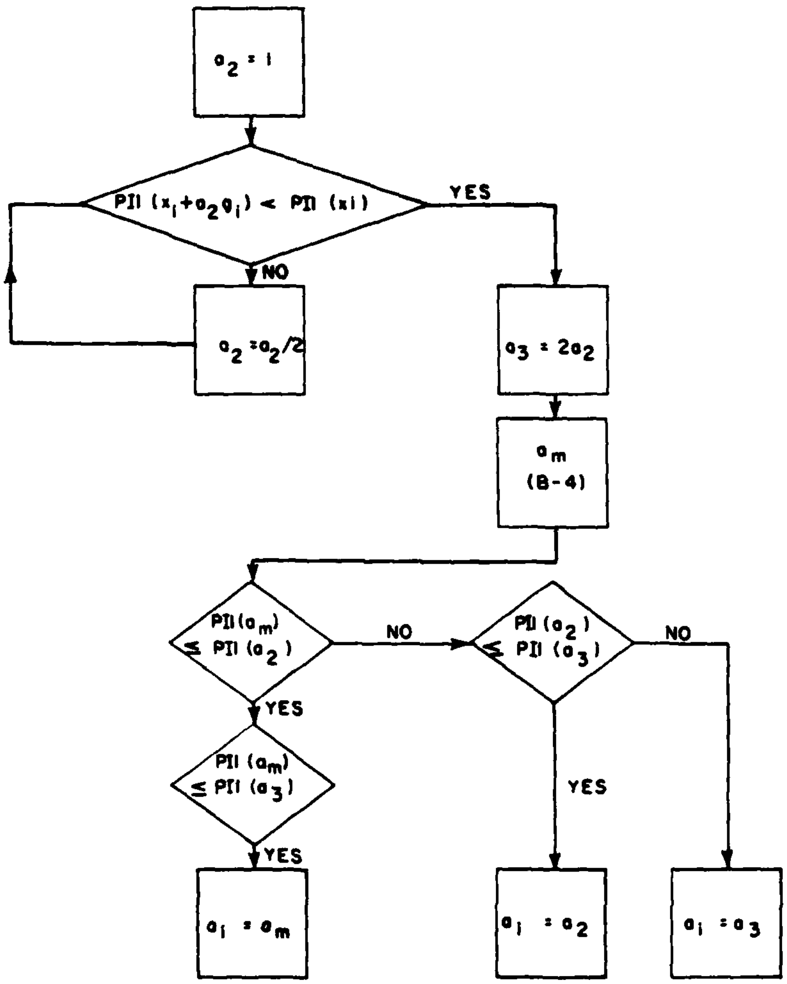


Figure B-1. A Diagram for Selecting a_i, the Step Size for the ith Iteration of the Modified Gauss-Newton Method

APPENDIX C. NUMBER OF OPERATIONS FOR THE MODIFIED GAUSS-NEWTON METHOD

The number of operations (additions, multiplications, divisions, square roots, and trigonometric functions) required for one iteration of the modified Gauss-Newton method is determined in this appendix.

This method is defined by

$$x_{i+1} = x_i + a_i \; g_i \tag{C-1}$$

where a_i is the step size and g_i is determined from

$$J(x_i) \; g_i = f(x_i) \tag{C-2}$$

by applying a Householder transformation to the $(J(x_i) \mid f(x_i))$ array. This transformation gives

$$R \; g_i = \bar{f} \tag{C-3}$$

where R is an upper triangular matrix and g_i is found by back substitution.

The number of operations needed to determine a_i is given, primarily, by the number of additional times that PI1 must be evaluated. Appendix B showed that a minimum of two additional evaluations per iteration are required, and the experiments demonstrated that frequently this was all that was needed. Furthermore, equation (B-4) shows that in addition to these operations, 5 additions, 6 multiplications, and 1 division are required. The divisions needed to reduce the step size (see figure B-1), assuming that a reduction is called for, are not counted. Therefore, the minimum number of operations required to find the step size are 2(K-1)+5 additions, 2 K+6 multiplications, and 1 division.

The equations which define the elements in the kth row of the $(J(x_i) \,\vdots\, f(x_i))$ array are given in appendix A. The operations needed to evaluate these elements are shown in table C-1. The operations needed for PI1 are included.

Table C-1. Number of Operations Required for the $(J\vdots f)$ Array and PI1.

Equation	+	x	÷	$\sqrt{\ }$	trig
$f(k) = R_{11}(k) - R_s(k)t_1(k)c/b$	1	2	-	-	-
$R_{11}(k) = -R_x(k)\sin\psi(k) - R_y(k)\cos\psi(k)$	1	2	-	-	2*
$R_s(k) = (R_x{}^2(k) + R_y{}^2(k) + R_z{}^2(0))^{1/2}$	2	3	-	1	-
$\partial f/\partial R_{xt}(0) = -\sin\psi(k) - R_x(k)t_1(k)c/R_s(k)b$	1	2	1	-	**
$\partial f/\partial R_{yt}(0) = -\cos\psi(k) - R_y(k)t_1(k)c/R_s(k)b$	1	1	**	-	**
$\partial f/\partial R_{zt}(0) = - R_z(0)t_1(k)c/R_s(k)b$	-	1	**	-	-
$\partial f/\partial V_{xt} = kh\ \partial f/\partial\ R_{xt}(0)$	-	2	-	-	-
$\partial f/\partial V_{yt} = kh\ \partial f/\partial\ R_{yt}(0)$	-	1	-	-	-
Total for one row	6	14	1	1	-
Total for K rows	6K	14K	K	K	*
PI1 requires	K-1	K	-	-	-

*Only 2 per leg, so they are not included in the totals.

**c/b is computed off-line, so $c/r_s(k)b$ is 1 division. Also $t_1(k)c/R_s(k)b$ is computed only once.

The number of operations needed to find $R_x(k)$ and $R_y(k)$, given an estimate of the target's velocity and initial position and the observer track, are not counted because they are the same for all iterative estimation algorithms which use a Cartesian signal model.

The Householder transformation used in the iterative least-squares algorithm is, in fact, a product of Householder transformations. The first member of this product operates on the matrix

$$A = (J \mid f), \tag{C-4}$$

which has K rows and n columns, where $n = 6$. The second member in this product leaves the first row and column of the transformed A matrix unchanged, so it operates on a (K-1) by (n-1) matrix. The next member operates on a (K-2) by (n-2) matrix and so on until the first (n-1) columns of A are transformed. The number of operations needed to transform A so that its new first column has the required structure is shown in the following table. The symbols used in this table are consistent with the Householder transformation algorithm shown on page 63 of reference 3, except that K is used here instead of m.

Table C-2. Number of Operations Required to Transform the First Column of the A Matrix

Symbol	+	x	÷	$\sqrt{\ }$
s	K-1	K-1	-	1
u(1)	1	-	-	-
β	-	1	1	-
γ	(K-1) (n-1)	(K+1) (n-1)	-	-
$\tilde{A}$	K(n-1)	K(n-1)	-	-
Total	Kn+(K-1)(n-1)	2Kn+n-K+1	1	1

In the omnidirectional least-squares problem, n = 6, hence, the first five columns of A must be transformed to upper triangular form. The next table shows the number of operations needed for each of these transformations.

Table C-3. Number of Operations Required for the Householder Transformation of A

Transformation Defined by Column No.	+	x	÷	$\sqrt{\ }$
1	Kn+(K-1)(n-1)	2Kn+n-K+1	1	1
2	(K-1) (n-1)+(K-2) (n-2)	2(K-1) (n-1)+n-K+1	1	1
3	(K-2) (n-2)+(K-3) (n-3)	2(K-2) (n-2)+n-K+1	1	1
4	(K-3) (n-3)+(K-4) (n-4)	2(K-3) (n-3)+n-K+1	1	1
5	(K-4) (n-4)+(K-5) (n-5)	2(K-4) (n-4)+n-K+1	1	1
Total	10Kn-25K-25n+85	10Kn-25K-15n+65	5	5
Total with n = 6	35K-65	35K-25	5	5

To compute PI2 requires (K-6) additions and (K-5) multiplications.

The back substitution algorithm requires 10 addtions, 10 multiplications, and 5 divisions.

The total number of operations required for one iteration of the modified Gauss-Newton method is shown in table C-4.

Table C-4. Final Count of Operations for One Iterations of the Modified Gauss-Newton Method

Function	+	x	÷	$\sqrt{\ }$
Step size	2K+3*	2K+6*	1*	-
(J ¦ f)	6K	14K	K	K
PI1	K-1	35K-25	-	-

*minimum

Table C-4. Final Count of Operations for One Iterations of the Modified Gauss-Newton Method (Cont'd)

Function	+	x	÷	√
Householder Transformation	35K-65	35K-25	5	5
PI2	K-6	K-5	-	-
Back substitution	10	10	5	-
Gauss-Newton Total	43K-62	51K-20	K+10	K+5
Minimum Number of Extra Operations for Modified G-N	2K+3	2K+6	1	0

When faced with an ill-conditioned problem, it is advisable to interchange the columns (and maybe the rows as well) of J before applying the Householder transformation. This is explained in reference 6. The number of operations required for this interchange are now shown in table C-4.

APPENDIX D. COMPUTER PROGRAM LISTING

This appendix contains the computer programs used to generate the results shown in section 4. There are two parts to this program: the first generates the averaged measurements; the second estimates the target parameters. Six significant digits are carried in all computations.

```
0 LPRINT"THIS PROGRAM GENERATES THE AVERAGED MEASUREMENTS"
1 DATA 10,10,10,0,90,0,-10,0,10,-5,10,10000,10000,1000,0,0
5 '--------------------------------------------------------------
6 ' TRANSLATION OF THE NAMES OF THE VARIABLES USED IN THIS PROGRAM
7 '--------------------------------------------------------------
10 'SO=SPEED OF THE OBSERVER. INPUTTED AS S0,S1;S2.
11 'GAMAC=HEADING OF THE OBSERVER. INPUTTED AS G0,G1,G2
12 'VZO=Z VELOCITY OF THE OBSERVER. INPUTTED AS Z00,Z10,Z20
13 'VXT=X VELOCITY OF THE TARGET
14 'VYT=Y " "       "          "
15 'RXT=X POSITION OF THE TARGET
16 'RYT=Y     "               "
17 'RZT=Z     "               "
18 'V1XO=X VELOCITY OF THE OBSERVER
19 'V2YO=Y        "           "
20 'R1XO=X POSITION OF THE OBSERVER
21 'R2YO=Y      "            "
22 'R3ZO=Z      "            "
23 'R4X=RELATIVE X POSITION(TARGET-OBSERVER)
24 'R5Y=     "     Y      "             "
25 'R6Z=     "     Z      "             "
26 'MEAN=MEAN OF MEASUREMENT NOISE
27 'SDEV=STANDARD DEVIATION OF THE MEASUREMENT NOISE
28 'C11=GAMAC-180 IN RADIANS. ANGLE OF ROTATION
29 'H=SAMPLING PERIOD
30 'RA=R11,PARALLEL TO CENTER LINE OF ARRAY
31 'RP=RT,PERPENDICULAR          "
32 'ALPHA=COS(TRUE ANGLE)
33 'TDELAY=TRUE TIME DELAY
34 'BI=USED IN RANDOM NUMBER GENERATOR
35 'NOISE=  "       "          "
36 'TMEAS=NOISY TIME DELAY
37 'AVTM=ONE DIMENSIONAL VARIABLE ,36 ELEMENTS.AVERAGED TMEAS
38 'STMEAS=USED TO GENERATOR AVTM
100 '*** INPUT OBSERVER INITIAL PARAMETERS AND MANEUVERS. INPUT
        TARGET PARAMETERS. INPUT MEAN AND SDEV OF MEASUREMENT NOISE.***
103 '
104 '
105 PRINT"DATA CARD 1,2,ETC.--OBSERVER SPEED,0,1,2--OBS. HEADING,0,1,2----Z VELO
CITY 0,1,2------TARGET X VEL.,Y VEL.,X POS.,Y POS.,Z POS.,------NOISE MEAN,SDEV-
-****** TYPE CONT AFTER DATA IS    ENTERED"
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
108 '
109 '
110 '*** STOP STATMENT ALLOWS USER TO ENTER DATA. TYPE CONT
        TO CONTINE ONCE DATA IS ENTERED ***
111 STOP
117 '------------------------------------------------------------
118 '
120 '*** READS INPUT DATA DEFINED IN REM 100.***
     *** INITIAL OBSERVER SPEED AND HEADING (IN DEGREES
         AND RADIANS). ***
125 READ S0,S1,S2,G0,G1,G2,Z0C,Z1C,Z2C,VXT,VYT,RXT,RYT,RZT,MEAN,SDEV
130 S0=S0:GAMA0=G0:HH=G0*.0174533
131 '
132 '
133 '*** SAMPLING PERIOD  IS H SECONDS.***
134 H=1
135 '
136 '             RANDOM IS USED BY THE NOISE GENERATOR
137 RANDOM
138 '           *** INITIAL X,Y,AND Z OBSERVER SPEEDS ARE SET. ***
140 V1X0=S0*SIN(HH):V2Y0=S0*COS(HH):VZ0=Z0C
142 '
143  *** ELIMINATES INITIAL OOFSET IN LINE 320.***
145 R1X0=-V1X0*H:R2Y0=-V2Y0*H:R3Z0=-VZ0*H
148 '------------------------------------------------------------
149 '            INPUT FOR AVERAGED TMEAS LOOP WHICH STARTS AT 700
150 DIM AVTM(35) :STMEAS=0 :L=0 :N=0
151 '            THERE ARE 36 ELEMENTS IN AVTM. 0-ELEMENT CORRESPONDING WITH
                        TIME=9.5 SECONDS AND 36-ELEMENT WITH 709.5 SEC.
152 '
153 '*** ELIMINATES INITIAL OFFSET IN LINE 340.***
155 RXT=-VXT*H+RXT:RYT=-VYT*H+RYT
158 '
159 LPRINT "K";"       ";"RAX";"        ";"REY";"        ";"R6Z","RA";"      ";"R?";
"      ";"TDELAY";"TMEAS"
160 '
161 '
162 '
163 '------------------------------------------------------------
195 '*** START OF MAIN LOOP. RUNS THREE LEGS, 4 MINUTES EACH.***
196 '
197 '            STEP ON COUNTER K IS H, THE SAMPLING PERIOD
198 '------------------------------------------------------------
200 FOR K=0 TO 720 STEP 1
238 '*** CHECKS FOR FIRST OBSERVER MANUVER AND SET
         NEW SPEED AND HEADING.***
239 '
240 IF 240<=K AND K<480 THEN 245 ELSE 260
245 S0=S1:GAMA0=G1:VZ0=Z1C
255 '
258 '*** CHECKS FOR SECOND OBSERVER MANUVER AND SETS
         NEW SPEED AND HEADING.***
260 IF K>=480 THEN S0=S2: GAMA0=G2:VZ0=Z2C
272 '
278 '*** FINDS ROTATION ANGLE FOR THE ? AXIS ROTATION ***
     *** FINDS OBSERVER HEADING IN RADIANS ***
280 C11=(GAMA0-180)*.0174533:GAMA0=GAMA0*.0174533
```

```
295 '
298 '*** X,Y,AND Z VEL. AND POSITION OF THE OBSERVER.***
     ****************************************************
     ****************************************************
300 V1XO=SO*SIN(GAMAO):V2YO=SO*COS(GAMAO)
320 R1XO=V1XO*H+R1XO:R2YO=V2YO*H+R2YO:R3ZO=VZO*H+R3ZO
330 '
338 '*** X,Y,AND Z TARGET POSITION.***
     *****************************
     *****************************
340 RXT=VXT*H+RXT:RYT=VYT*H+RYT:RZT=RZT
358 '*** RELATIVE X,Y,AND Z POSITION (TARGET-OBSERVER).***
                 *************************
                 *************************
360 R4X=RXT-R1XO:R5Y=RYT-R2YO:R6Z=RZT-R3ZO
361 '
362 '***          RELATIVE X,Y,Z ARE FORMED.        ***
     ***          TRANSFORMATION TO TOWED           ***
     ***          ARRAY COORDINATES IS NEXT.        ***
365
366 '
367 '---------------------------------------------------------
500 '*            TRASFORMATION            *
520 RA=R4X*SIN(C11)-R5Y*COS(C11):  RP=R4X*COS(C11)-R5Y*SIN(C11)
540 RS=(RA*RA+RP*RP+R6Z*R6Z)^.5
550 '---------------------------------------------------------
552 '             *** TRUE TIME DELAY ***
560 ALPHA=RA/RS
580 TDELAY=ALPHA/4000.
581 '---------------------------------------------------------
600 '
620 '***    GAUSSIAN NOISE GENERATOR.    ***
630 '
635 '             12 NUMBERS FROM UNIFORM(0 TO 1)
640 BI=-6
650 FOR I=1 TO 12
660    BI=BI+RND(0)
670 NEXT I
674 '
675 '             NOISE=N(MEAN,SDEV^2)
680 NOISE=SDEV*BI+MEAN
685 '---------------------------------------------------------
686 '
687 '             NOISY TIME DELAY MEASUREMENT
690 TMEAS=TDELAY+NOISE
691 '---------------------------------------------------------
692 IFL=0 THEN 694 ELSE 700
694 LPRINT K;R4X;R5Y;R6Z,RA;RP;TDELAY;TMEAS
700 '--------------------------------------------------------
701 '
702 '
703 '    AVERAGED TMEAS AT 9.5,29.5,49.5,--------,709.5 SEC.
704
710 STMEAS=STMEAS+TMEAS
711 L=L+1
718 '            COMPLTES AND STORES AVERAGED TMEAS AND RESETS
             L,STMEAS,AND INCREMENTS N
719 '
720 IF L=20 THEN L=0:AVTM(N)=STMEAS/20. :N=N+1 :STMEAS=0
800 NEXT K
900
```

```
801 '-----------------------------------------------------------
802 PRINT"AVERAGED TMEAS IS LISTED BELOW. THERE ARE 36 ELEMENTS 0-TO-35,
          INCREASING FROM LEFT TO RIGHT."
803 '
850 FOR N=0 TO 35
860   LPRINT AVTM(N);
870 NEXT N
880 '-----------------------------------------------------------
1000 '----------------------------------------------------------
1900 '
1910 '
1920 '          *** WRITE AVERAGED TMEAS TO DISK ***
2000 OPEN"O",1,"TMEAS.DAT"
2100 FOR I=0TO35
2200 PRINT#1,AVTM(I)
2300 NEXTI
2400 CLOSE
2410 '----------------------------------------------------------
3000 FOR I=1 TO 3
3100 LPRINT CHR$(138);NEXT
3200 END
```

```
0 LPRINT"ESTIMATE TARGET PARAMETERS--MODIFIED GAUSS-NEWTON METHOD"
1 DATA10,10,10,0,90,0,0,0,0
10 '-----------------------------------------------------------
20    GIVEN THE ESTIMATE OF THE TARGET PARAMETERS, THIS PROGRAM
      GENERATES THE JACOBIAN AND THE FORCING FUNCTION.
30 '-----------------------------------------------------------
31 '
32
35 '     TRANSLATION OF THE NAMES OF THE VARIABLES
36 '-----------------------------------------------------------
37 'RS=SLANT RANGE
38 'M1=DUMMY VARIABLE USED IN LINE 1670
40 'AVTM=AVERAGED TIME DELAY MEASUREMENTS
41 'LL=ITERATIVE COUNTER
42 'W1,W2,& W3 STORE INITIAL X,Y,&Z TARGET POSITION
43 'A=JACOBIAN WITH -F(X) IN THE SIXTH COLUMN
44 'C11=TRANSFORMATION ANGLE
45  P1I=PERFORMANCE INDEX BEFORE HOUSEHOLDER
46 'P2I=  "          "    AFTER        "
47  U=USED IN HOUSEHOLDER
48 'SIGN=  "      "
49 'MAG=    "     "
50 'BATA=  "      "
51  ETA=   "     "
52 'X= CORRECTION TO TARGET PARAMETERS
53 'B= USED IN BACK SUBSTITUTION
54 'F1 AND F2 ARE FIRST AND LAST MEASUREMENT WEIGHTS, RESPECTIVELY
55 'TT=TEMPORARY STORAGE
56 'T(35,5)= TEMPORARY STORAGE FOR INTERCHANGE OF ROWS OF A
57 'KI=USED IN QUADRATIC FIT
58 ' ZI=USED IN QUADRATIC FIT
59 ' IPP(KI)= "      "       "           "
60 ' SA= "      "            "           "
61 ' A2= "      "       "       "       "
62 ' Q1=TERMINATION PARAMETER
63 ' Q2=  "      "       "
64 ' Q5= OUTPUT PI TO DISK PARAMETER
99 '
100 '-----------------------------------------------------------
1000 '    THE FIRST FIVE COLUMNS OF A(35,5) CONTAIN THE JACOBIAN
        WHILE THE SIXTH (& LAST) COLUMN CONTAINS -F(X).
1010 '
1020 '    THE AVTM, AVERAGED TDELAY MEASUREMENTS, DATA IS CALLED
        FROM DISK.
1030 DIM AVTM(35),A(35,3),U(35),X(4),B(4),T(35,5)
1035 INPUT"FILESPEC";FILESPEC$
1040 OPEN "I",1,FILESPEC$
1050 FOR L=0 TO 35
1060 INPUT#1,AVTM(L)
1070 PRINT "L=";L;"AVTM=";AVTM(L);
1080 NEXT L
1090 CLOSE
1100 LPRINT"-------------------------------------------------------"
1105 LPRINT"FILESPEC = ";FILESPEC$
1107 LPRINT"-------------------------------------------------------"
1110 '-----------------------------------------------------------
```

```
1110 '   *** OPTION TO WEIGHT THE FIRST & LAST MEASUREMENTS ***
1120 INPUT "MEASUREMENT WEIGHTING";N$
1130 IF N$="NO" THEN GOTO 1160
1140 INPUT"SET FIRST AND LAST WEIGHTS";F1,F2
1150 LPRINT"FIRST AND LAST WEIGHTS ARE:";F1;"AND";F2
1160 '
1170 PRINT
1180 PRINT"*****************************************************"
1190 '
1200 '
1210 '-----------------------------------------------------------
1220 'READS INPUT DATA DEFINED IN 1900
1230 READ S0,S1,S2,G0,G1,G2,Z00,Z10,Z20
1240 '   *** SET INITIAL TARGET PARAMETERS ***
1250 '-----------------------------------------------------------
1260 LL=-1
1270 TT=-1
1280 P1I=1E+12
1290 VXT=0:VYT=0
1300 RS=20000
1310 RZT=3000
1320 RA=RS>400(KAVTH(0)
1330 RT=TT*(RS*RS-RA*RA-RZT*RZT)^.5
1340 C11=(G0-19.)*.0174533
1350 RXT=RT*COS(C11)+RA*SIN(C11)
1360 RYT=-RT*SIN(C11)+RA*COS(C11)
1365 GS=0
1370 '
1380 '            ***     LL=ITERATIVE COUNT    ***
1390 '-----------------------------------------------------------
1400 LL=LL+1
1401
1410 KI=0:ZI=0
1420 KI=KI+1
1425 IF KI=19 THEN PRINT"WRONG DIRECTION": END
1428 '
1430 LPRINT"##################################################"
1440 LPRINT"OBSERVER AND TARGET PARAMETERS","LL=";LL;"KI=";KI;"IPF(KI-1)=";IPF(K
I-1);"ZI=";ZI;"SA=";SA
1450 LPRINT"-----------------------------------------------------------"
1460 LPRINTS0,S1,S2,G0,G1,G2,Z00,Z10,Z20,VXT,VYT,RXT,RYT,RZT
1470 '            SETS OBSERVER INITIAL SPEED AND HEADING IN RADIANS
1480 LPRINT
1490 LPRINT"-----------------------------------------------------------"
1500 '            *** STORES TARGET RANGE COMPONENTS ***
1510 W1=RXT:W2=RYT:W3=RZT
1520 S0=S0        :GAMA0=G0*.0174533
1530 V1X0=S0*SIN(GAMA0) :V2Y0=S0*COS(GAMA0)      :VZ0=Z00
1540 'INITIAL POSITION OF THE OBSERVER,I.E.,POSITIN AT 9.5 SECONDS
1550 '
1560 R1X0=V1X0*9.5       :R2Y0=V2Y0*9.5  :R3Z0=VZ0*9.5
1570 'INITIAL POSITION OF THE TARGET(AT 9.5 SECONDS)
1580            *** STORE INITIAL TARGET POSITION ***
1590 RXT=VXT*9.5+RXT:RYT=VYT*9.5+RYT
1600 '-----------------------------------------------------------
1610 '          RELATIVE POSITION AT 9.5 SECONDS
1620 R4X=RXT-R1X0:R5Y=RYT-R2Y0:R6Z=RZT-R3Z0
1630 RI=(R4X*R4X+R5Y*R5Y+R6Z*R6Z)^.5
```

```
1640 '
1650 '-----------------------------------------------------------
1660 '       *** -F(0) AND J(0,1,2,3,4) AT TIME=9.5 SECONDS STORED
 IN A(0,I). F AND J DON'T APPEAR IN THE PROGRAM. ***
1670 C11=GAMAO-180*.0174533:RA=R4X*SIN(C11)+R5Y*COS(C11)
1680 '
1690 A(0,5)=-(RA-RS*4000*AVTM(0))
1700 A(0,4)=-4000*AVTM(0)*R6Z/RS:A(0,3)=COS(C11)-4000*AVTM(0)*R5Y/RS:A(0,2)=SIN(
C11)-4000*AVTM(0)*R4X/RS
1710 A(0,1)=9.5*A(0,3):A(0,0)=9.5*A(0,2)
1720 '-----------------------------------------------------------
1730 '
1740 P11=A(0,5)*A(0,5)
1750 '
1760 '           MAIN LOOP K=1(TIME=29.5) TO K=35 (TIME=709.5 SEC.)
1770 '           **********************************************
1780 '
1790 '
1800 FOR K=1 TO 35
1810 '           CHECKS FOR OBSERVER MANEUVERS
1820 IF K=12 THEN 1850 ELSE 1950
1830 ' COMPUTE POSITION AT 229 SECONDS, THEN UPDATE SO & GAMAO
1840 '
1850 R1XO=V1XO*9.5+R1XO :R2YO=V2YO*9.5+R2YO :R3ZO=VZO*9.5+R3ZO
1860 '
1870 SO=S1 :GAMAO=G1*.0174533 :VZO=Z10
1880 V1XO=SO*SIN(GAMAO) :V2YO=SO*COS(GAMAO)
1890 '
1900 '    ***COMPUTE OBSERVER POSITION AT K=12(TIME=249.5 SEC.) ***
1910 R1XO=V1XO*10.5+R1XO:R2YO=V2YO*10.5+R2YO:R3ZO=VZO*10.5+R3ZO:GOTO 2110
1920 '-----------------------------------------------------------
1930 '
1940 '           CHECKS FOR SECOND OBSERVER MANEUVER
1950 IF K=24 THEN 1970 ELSE 2070
1960 '           COMPUTE OBS. POS. AT 479 SEC. THEN UPDATE SO & GAMAO
1970 R1XO=V1XO*9.5+R1XO :R2YO=V2YO*9.5+R2YO :R3ZO=VZO*9.5+R3ZO
1980 '
1990 SO=S2 :GAMAO=G2*.0174533 :VZO=Z20
2000 V1XO=SO*SIN(GAMAO) :V2YO=SO*COS(GAMAO)
2010 '           COMPUTE OBS. POS. AT K=24(TIME=489.5 SEC.)
2020 '
2030 R1XO=V1XO*10.5+R1XO:R2YO=V2YO*10.5+R2YO:R3ZO=VZO*10.5+R3ZO
2040 GOTO 2110
2050 '           IF A MANEUVER OCCURS THE NEXT POSITION IS COMPUTED
             ABOVE , SO THE LOGIC SHOWN BELOW IS SKIPPED.
2060 '-----------------------------------------------------------
2070 '  X,Y, AND Z POSITION OF THE OBSERVER
2080 '
2090 R1XO=V1XO*20+R1XO        :R2YO=V2YO*20+R2YO    :R3ZO=VZO*20+R3ZO
2100 '-----------------------------------------------------------
2110 '
2120 'ESTIMATE OF X,Y, AND Z POSITION OF THE TARGET
2130 RXT=VXT*20+RXT   :RYT=VYT*20+RYT:RZT=RZT
2140 '  *** ESTIMATE OF RELATIVE POSITION ***
2150 '
2160 R4X=RXT-R1XO:R5Y=RYT-R2YO:R6Z=RZT-R3O
2170 RS=(R4X*R4X+R5Y*R5Y+R6Z*R6Z)^.5
2180 '-----------------------------------------------------------
2190 '  *** ELEMENTS OF THE F-ARRAY STORED AS -F IN A(K,5) *** ***
```

```
2200 '
2210 C11=(AMAI-180)*.0174533
2220 RA=R-X*SIN(C11)+RSY*COS(C11)
2230 M1=4000*A/TM(K)
2240 A(K,5)=-(RA-RS*M1)
2250 '
2260 '  *** ELEMENTS OF THE J-ARRAY ***
2270 '-----------------------------------------------------------
2280 A(K,4)=-M1*R6Z/RS
2290 A(K,3)=COS(C11)-M1*REY/RS
2300 A(K,2)=SIN(C11)-M1*R4X/RS
2310 A(K,1)=(K*ZI+9.5)*A(K,3)
2320 A(K,0)=(K*ZI-9.5)*A(K,2)
2330 '
2340 P1I=P1I+A(K,5)*A(K,5)
2350 '-----------------------------------------------------------
2361 NEXT K
2362 '----------------------------------------------------------
2363 '  *** OUTPUT PI TO DISK ***
2364 IF QS=1 THEN 5000
2365 '----------------------------------------------------------
2366 IF LL=0 THEN 2375
2367 '  *** TERMINATE BY INSERTING TRUE TARGET PARAMETERS ***
2368 Q1=(P1I-P2I)/P2I  :Q2=ABS(Q1)
2369 IF Q2 =0.1 THEN 4374
2370 '----------------------------------------------------------
2374 '
2375 '  *** START OF QUADRATIC FIT ***
2380 IPP(KI)=P1I
2385 IF LL=0 THEN 2571
2390 IF ZI=2 THEN 2570
2400 IF ZI=1 THEN 2530
2410 IF IPP(KI) =PI(LL-1) THEN 2450
2420 SA=-1/(ZI KI)
2430 GOSUB 4450
2440 GOTO 1420
2450 IF KI 1 THEN 2490
2460 SA=1:ZI=1
2470 GOSUB 4450
2480 GOTO 1420
2490 A2=(3*PI(LL-1)-4*IPP(KI)+IPP(KI-1))/(2*PI(LL-1)-4*IPP(KI)+2*IPP(KI-1)):ZI=2
2500 SA=(A2-1)/(ZI(KI-1))
2510 GOSUB 4450
2520 GOTO 1420
2530 IF IPP(KI) IPP(KI-1) THEN 2571
2540 SA=(-PI(LL-1)+4*IPP(KI-1)-3*IPP(KI))/(2*(PI(LL-1)-2*IPP(KI-1)+IPP(KI))):ZI=
2
2550 GOSUB 4450
2560 GOTO 1420
2570 PI(LL)=IPP(KI)
2575 '  *** END OF QUADRATIC FIT ***
2580 LPRINT'----------------------------------------------------------'
2585 LPRINT"LL=";LL;"PI(LL)=";PI(LL)
2590 ... 
2595 IF LL= THEN ...
2597 IF LL= THEN ...
2600 '-----------------------------------------------------------
2605 '  *** WEIGHTING OF FIRST & LAST MEASUREMENTS ***
```

```
2610 IF N$="NO"THEN GOTO 2710
2620 LPRINT"A(0,5)=";A(0,5);"A(35,5)=";A(35,5)
2630 FOR I=0 TO 5
2640           A(0,I)=F1*A(0,I)
2650           A(35,I)=F2*A(35,I)
2660 NEXT I
2669 '-----------------------------------------------------------
2665 '   *** BECAUSE THE FIRST & LAST ROWS HAVE THE LARGEST ELEMENTS
        WHEN WEIGHTED, THEY ARE STACKED TOGETHER ***
2670 'INTERCHANGE THE 0TH AND THE 34TH ROWS OF A(K,J)
2680 FOR J=0 TO 5
2690   TT=A(34,J):A(34,J)=A(0,J):A(0,J)=TT
2700 NEXTJ
2702 '-----------------------------------------------------------
2705 '   *** BECAUSE THE LAST ROWS OF MATRIX A HAVE THE LARGEST ELEMENTS
                THEY ARE STACKED FIRST FOR THE HOUSEHOLDER TRANSFORMATION ***
2710 'INTERCHANGE THE ROWS OF A
2720 GOSUB 3860
2724 '
2725 '   *** CALL HOUSEHOLDER ***
2730 GOSUB 3230
2840 '
2850 '
2860 PRINT"##########################################################"
2870 '
2880 '
2890 '*** BACK SUBSTITUTION IS NEXT ***
2900 '
2910    FOR I=0 TO 4
2920            B(I)=A(I,5)
2930    NEXT I
2940
2950    FOR L=4 TO 0 STEP -1
2960            X(L)=B(L)/A(L,L)
2970            FOR I=0 TO L-1
2980                    B(I)=B(I)-A(I,L)*X(L)
2990            NEXT I
3000 NEXT L
3010 '
3070 '-------------------------------------------------------------
3080 '          *** UPDATE TARGET PARAMETERS ***
3090 '
3100 VXT=VXT+X(0)     :VYT=VYT+X(1)
3120 RXT=W1+X(2)      :RYT=W2+X(3)       :RZT=W3+X(4)
3140 LPRINT"------------------------------------------------------------"
3150 LPRINT"LL=";LL;"   X(0) THRU X(4)     ";X(0);X(1);X(2);X(3);X(4)
3160 LPRINT"-----------------------------------------------------------"
3170 '@@@@@@@@@@    END OF MAIN LOOP       @@@@@@@@@@
3180 GOTO 1400
3190
3200
3210 '=============================================================
3220 '  *** SUBROUTINE HOUSEHOLDER ***
3230 '   *** HOUSEHOLDER TRANSFORMATION ***
3290 '
3300 '          REQUIRED INPUT ,COLS &ROWS( COLS=PRINT,ROWS=PRINT)
3310 '                  NOT REQUIRED IF CONTAINED IN MAIN PROGRAM
```

```
3320 '-------------------------------------------------------
3330 '                   SET THE NUMBER OF ROWS,COLS
3340 N=COLS-1    :M=ROWS-1
3350 '           *** MAIN LOOP: TRANSFORMS A COLUMN AT A TIME.
                     STOPS AT COL-(N-1)--ASSUMES LAST COL IS
                     FORCING FUNCTION.  ***
3360
3370 FOR K=0 TO (N-1)
3380             MAG=0
3390 '           *** COMPUTES MAGNITUDE OF COL K
3400             FOR I=K TO M
3410                     IF K=M THEN GOTO 3860
3420 '                           ***     THE ABOVE CHECKS FOR SQ. ARRAY***
3430                                     MAG=MAG+A(I,K)*A(I,K)
3440             NEXT I
3450 '                   *** SETS NEW DIAGONAL ELEMENT
3460             SIGN=A(K,K)/ABS(A(K,K))
3470             MAG=-SIGN*(MAG^.5)
3480 '
3490 '                   COMPUTES U VECTOR
3500             U(K)=A(K,K)-MAG
3510             FOR J=K+1 TO M
3520                                     U(J)=A(J,K)    :A(J,K)=0
3530             NEXT J
3540 '                   *** K-TH COL IS UPDATED STARTING
                             AT THE K-TH ROW. ROW ELEMENTS
                             FROM K+1 ON ARE SET TO 0(SEE 2180)
3550             A(K,K)=MAG
3560                     *** THE REMAINING K-1 COLS ARE NOW TRANSFORMED
3570 '-------------------------------------------------------
3580             BATA=1/(MAG*U(K))
3590             FOR J=K+1 TO N
3600                     ETA=0
3610                     FOR I=K TO M
3620                             ETA=ETA+U(I)*A(I,J)
3630                     NEXT I
3640                     ETA=ETA*BATA
3650 '
3660 '           *** ELEMENTS OF THE I-TH COL OF A ARE REDEFINED.
3670 '-------------------------------------------------------
3680 '
3690                     FOR I=K TO M
3700                             A(I,J)=A(I,J)+ETA*U(I)
3710                     NEXT I
3720  *** MOVING TO THE NEXT COLUMN ***
3730             NEXT J
3740
3750 '*** MOVING TO THE NEXT TRANSFORMATION ***
3755 NEXT K
3760 '           *** FOR. INDEX THOW SQ.ERR AFTER TRANSFORMATION ***
3770 SI=0
3780 FOR I=N TO M
3790          SI=SI+A(I,N)*A(I,N)
3800 NEXT I
3810 LPRINT"*************************************************"
```

```
3830 '
3840 LPRINT"P2I=";P2I;"LL=";LL
3850 '
3860 LPRINT"****************************************************************"
3870 RETURN
3872
3876 '
3880 'SUBROUTINE FOR INTERCHANGING ALL ROWS OF A(35,5). RETURNS TO 2350
3890 FOR K=0 TO 33
3900    FOR J=0 TO 5
3910            T(K,J)=A(K,J)
3920    NEXT J
3930 NEXT K
3940 FOR K=0 TO 35
3950    FOR J=0 TO 5
3960            A(35-K,J)=T(K,J)
3970    NEXT J
3980 NEXT K
3990 RETURN
4300 '------------------------------------------------------------
4370 '----------------------------------------------------------
4373 '  *** FROM A GOTO IN LINE 2369 ***
4374 PI(LL)=P1I
4375 LPRINT"LL=";LL,"PI(LL)=";P1I,"Q2=";Q2;"P2I=";P2I
4378 Q5=1
4380 INPUT"VXT=";VXT
4390 INPUT "VYT=";VYT
4400 INPUT "RXT=";RXT
4410 INPUT "RYT=";RYT
4420 INPUT "RZT=";RZT
4430 GOTO 1400
4445 '----------------------------------------------------------
4450 '  *** SUBROUTINE TO UPDATE TARGET PARAMETERS ***
4460 '       CALLED ( LINES 2430,2470,2510,&2550 ) BY QUADRATIC FIT
4470 VXT=VXT+SA*X(0) :VYT=VYT+SA*X(1) :RXT=W1+SA*X(2) :RYT=W2+SA*X(3) :RZT=W3+SA
*X(4)
4480 RETURN
4490 '----------------------------------------------------------
5000 '            OUTPUT TO DISK
5001 '  *** FROM A GOTO IN LINE 2364 ***
5005 PI(LL)=P1I
5006 LPRINT "LL=";LL,"PI(LL)=";PI(LL)
5010 OPEN"O",2,"PI(LL)/D11:1"
5020 FOR LL=0 TO 9
5030 PRINT#2,PI(LL)
5040 NEXT LL
5050 CLOSE
5060 END
```

INITIAL DISTRIBUTION LIST

Addressee	No. of Copies
NAVSEA (SEA-63, -63R, -63R-1, -63R-13, -63X, -631X -63D (2 cy), PMS-409 (2 cy))	10
DTIC, Alexandria	12

END

DATE
FILMED
12-81

DTIC